

HLS

6th exercise set

Άσκηση 2^η

Ζητούμενο 1

```
COL:for (j = 0; j <M; ++j) {  
    // get values of the pixels in the kernel  
    p1 = (j > 1) ? img[i][j-2]: 0;  
    p2 = (j > 0) ? img[i][j-1]: 0;  
    p3 = img[i][j];  
    p4 = (j < M-1) ? img[i][j+1]: 0 ;  
    p5 = (j < M-2) ? img[i][j+2]: 0 ;  
    // compute the mean  
    out[i][j] = (p1 + p2 + p3 + p4 + p5) / 5;  
}
```

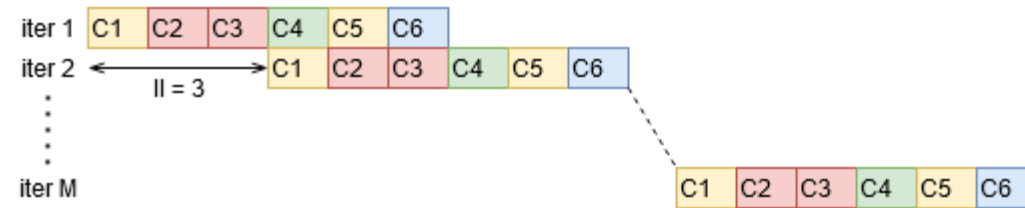
N rows, M columns

2 memory accesses per clock per memory

each col iter
worst case

- 5 memory reads
- 4 additions
- 1 division
- 1 memory write

after schedule we have:



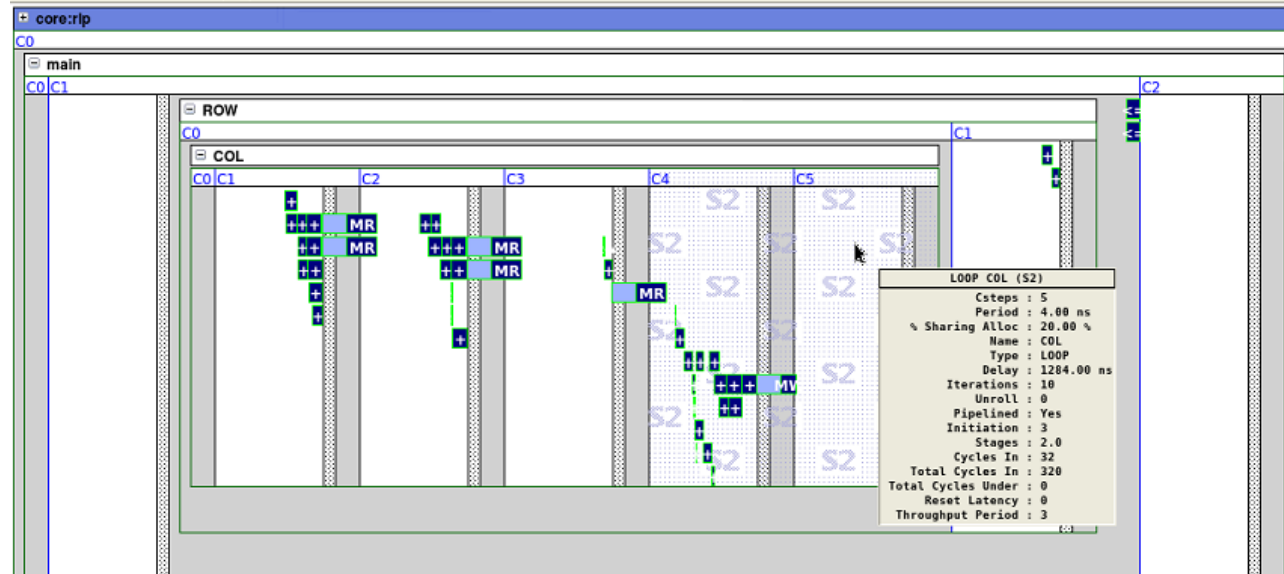
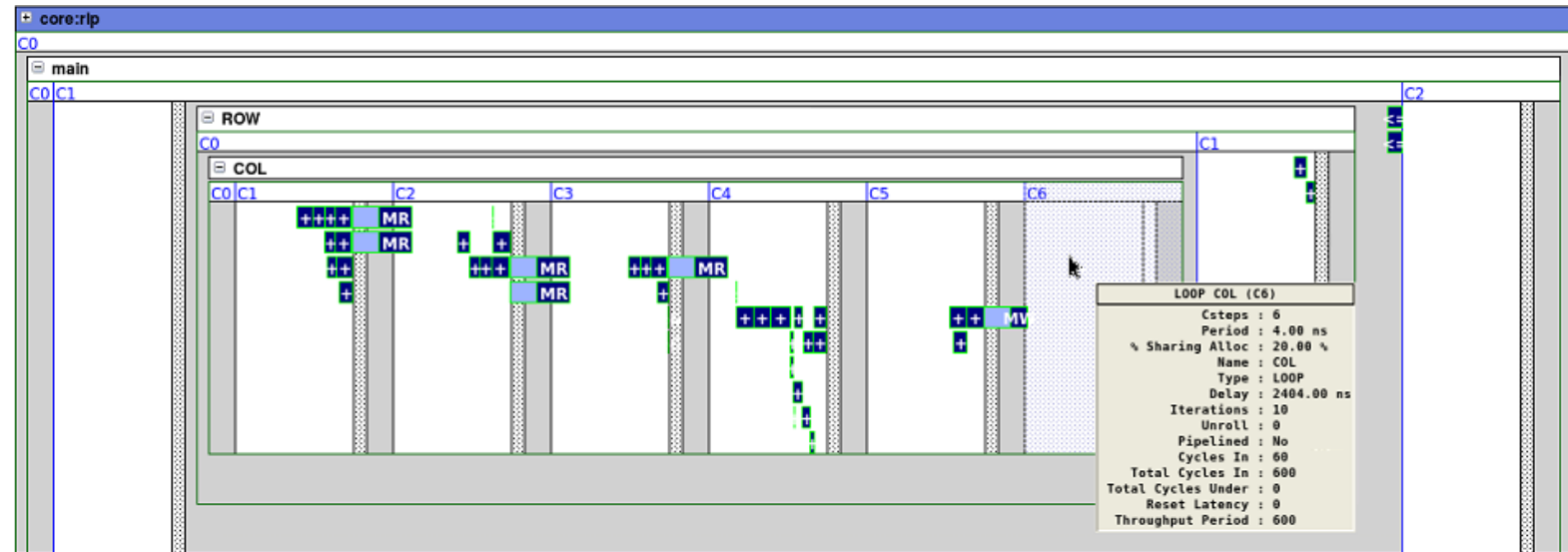
Παρατηρούμε ότι δεν υπάρχουν εξαρτήσεις ανάδρασης. Σε καμία επανάληψη δεν είναι απαραίτητο να ξέρουμε το αποτέλεσμα της προηγούμενης, αφού διαβάζουμε και γράφουμε σε διαφορετικές μνήμες. Το ελάχιστο ll είναι ίσο με 3 όπως φαίνεται στο σχήμα αφού για μικρότερο από αυτό θα πρέπει οι δουλειές των κύκλων C2 και C4 να εκτελεστούν παράλληλα το οποίο υποδηλώνει 3 προσπελάσεις τις μνήμης της αρχικής εικόνας στον ίδιο κύκλο, πράγμα αδύνατο.

Περιοριστικός παράγοντας: προσπελάσεις μνήμης/ IO dependency

Άσκηση 2^η

Ζητούμενο 1

Schedule πριν και μετά:



Άσκηση 2^η

Ζητούμενο 2

```
void filter(dtype img[N][M], dtype out[N][M]){
    dtype kernel[5];
    int k;
    ROW: for (int i = 0; i < N; ++i) {
        k = 0;
        kernel[0] = 0;
        kernel[1] = 0;
        kernel[2] = img[i][0];
        kernel[3] = img[i][1];
        kernel[4] = img[i][2];

        out[i][0] = (kernel[2] + kernel[3] + kernel[4]) / 5;

        COL: for (int j = 1; j < M; j++) {
            kernel[k] = (j >= M-2) ? 0 : img[i][j+2];

            // compute the mean
            out[i][j] = (kernel[0] + kernel[1] + kernel[2] + kernel[3] + kernel[4]) / 5;

            // circular buffer implementation
            k = (k==4) ? 0 : k+1;
        }
    }
}
```

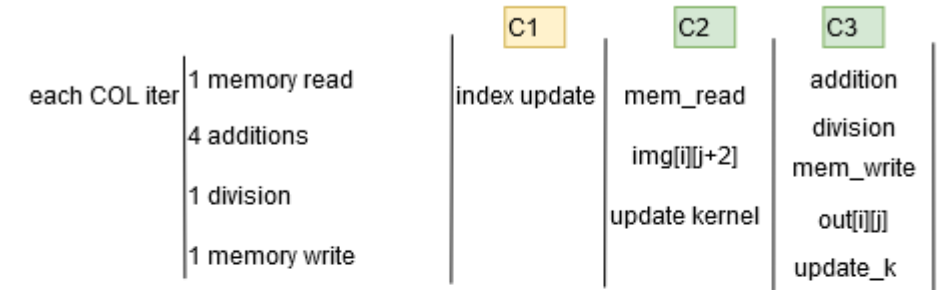
Υλοποιώ μια λειτουργία κυκλικού buffer.

Βγάζω εκτός του βρόγχου COL την πρώτη επανάληψη αφού έχει δουλειές που γίνονται μόνο σε αυτή. Το αποτέλεσμα είναι πλέον να μπορεί να εφαρμοστεί pipeline στον βρόγχο COL με $II = 1$

Σε κάθε επανάληψη γίνεται το πολύ μία προσπέλαση σε κάθε μνήμη, οπότε δεν υπάρχει περιορισμός στους πόρους. Μπορούν να γίνουν οι απαιτούμενες προσπελάσεις στον ίδιο κύκλο ρολογιού. Τα memory access δεν αποτελούν περιοριστικό παράγοντα

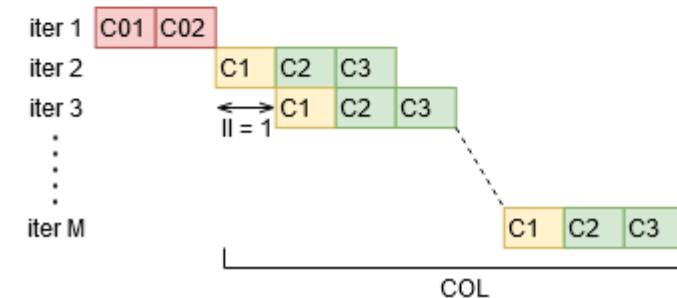
N rows, M columns

2 memory accesses per clock per memory valid



Παράδειγμα υπολογισμών. Οι προσβάσεις μνήμης μας ενδιαφέρουν που αποτυπώνονται όπως και στο schedule.

Η διαδικασία απο άποψη περιορισμών μνήμης μπορεί να εκτελεστεί και σε ένα κύκλο ρολογιού.



Άσκηση 2^η

Ζητούμενο 2

Schedule με pipeline $II=1$:

